FEB 1991   APR 1991

MAILBOX

Your space for comment and controversy. Write to
PC PLUS Mailbox, Beauford Court,
30 Monmouth Street, Bath, BA1 2BW

IN DEFENCE OF ADVERTISING

In response to Bill Lovett in December's *Mailbox*, I must say I find the adverts in PC PLUS very useful. I frequently buy computer equipment for my company, and I find them invaluable in finding not only the cheapest prices, but other deals such as on-site maintenance, bundled software and so on.

Being aware of the latest prices is vitally important in the computer arena; they move (generally downwards) so rapidly that I would estimate that I save the cost of an annual subscription to the magazine virtually every month by reading the latest dealer ads.

Philip D Naylor,
Purley on Thames,
Berkshire.

PC PLUS is proud of its record of editorial independence from commercial pressures, so we're always a little wary of making this type of point too enthusiastically. However it's true, so we will. Software and hardware prices in the PC market place are extremely 'elastic', and the only way to find out what something really costs is to see what the dealers are selling it for – and by far the most convenient place to find that information is in the dealer ads of computer magazines. For that reason we've always viewed the growth of 'dealer listings' adverts in PC PLUS as a real benefit to our readers, as well as (yes, we admit it) our coffers.

A BIT OF A BIND

A propos of Jaap van der Sijp's leter (*Mailbox*, December) on the demise of ring binders, I would like to suggest a reason even more charitable than your own for software manufacturers' unwillingness to provide them.

The problem lies in the metal rings found in most ring binders. All metal is to some extent magnetic, and even the small Tesla field emitted by a typical three-ring affair could be enough to suddenly wipe the smile off a Lotus issue disk or whatever. That must be it!

The answer of course is to use plastic rings – and please, suppliers, don't claim that it would be too expensive. If the price of software rises much further we'll expect to see it ready installed on a 486 PC, together with a pop-up manual on the disk as well.

Incidentally, I would have thought that the concepts of 'charity' and 'computing' are pretty well proven to be mutually exclusive by now. 'Faith', 'Hope' – yes, as a computer user you need these in plenty. But Charity? Not much chance really, is there?
Nigel Bourne,
London SW6.

TELECOM SCOLD

Tony Dennis's recent mention of British Telecom's Phonebase was unfortunately out of date even before it was published, B.T. having withdrawn the service until next April due to 'misuse of the system'. This is most mysterious – was the simultaneous introduction of the 43p charge for directory enquiries a coincidence?

I have invested money in purchasing a modem and software, and time in learning the system – including overcoming B.T.'s insistence on using an obscure terminal emulation that requires us to use strange [ALT] key combinations to get the input data accepted.

Phonebase should have offered significant savings, since the only charge is for standard call time. I will be interested to see in what form it re-appears.
Stephen Richards,
Tunbridge Wells,
Kent.

GETTING THE PICTURE

I can't agree with David Dala's view (January *Mailbox*) that the use of icons in computer software and elsewhere is a backward step, and likely to lead to widespread illiteracy.

Both icons and language have their places as methods of conveying information. While language is essential for communicating detailed facts and descriptions, icons are an extremely efficient way of putting quite large amounts of information in a form that is both compact and instantly recognisable.

One of the human brain's outstanding abilities is pattern recognition, still far ahead of anything achieved by a computer. A baby, unable to speak let alone read or write, can recognise its mother from a sea of basically similar human faces, while as adults we can recognise places and people from the patterns in images which contain only a small amount of actual picture information. For example when watching low-resolution domestic video recordings.

Far from being primitive, icons (or 'pictures', to give them their more traditional name) are an extremely powerful medium, which make the best use of our cognitive capabilities, and incidentally help to overcome the barriers of conventional language – as anyone who has gratefully recognised the 'his and hers' conveniences signs at a foreign airport will testify!
Keith Parry,
Norwich.

DISTORTIONATE PRICES

I have noticed that when you review a computer, you consider the amount of bundled software to be of great importance, because in your country software is considered to be too expensive. As a result of this I felt you might be interested in the conditions which prevail in my country, Pakistan, where, you may be surprised to hear, there is no copyright law at all.

This means that we can make as many copies of a software package as we like. In Pakistan the latest versions of major packages, such as *AutoCAD* and *Ventura Publisher,* are on sale for between ten and twenty pounds. Everything can be obtained for these prices – games, programming languages, the lot. Even if the software is copy protected, the protection can be removed.

So you can see why bundled software is unimportant to us, and why a lot of foreigners visit our software shops to buy at these prices. Incidentally, computer magazines are very popular here, with PC PLUS one of the favourites – and you can buy copies of the *SuperDisk* in the shops!

I hope you will print this letter, so that people in Britain will know what is happening in the Third World countries.
Asad Amin,
Karachi,
Pakistan.

Your letter leaves us just a bit confused about your personal attitude to all this software copying. It may be great fun ripping off copies of AutoCAD, but the Government of Pakistan (and quite a few others too) will have to do something about this illegal activity if their country and others is going to shake off that 'Third World' tag and join the Developed World.

MAILBOX SPECIAL – THE PASCAL DEBATE.

In an article in November's PC PLUS, Paul Stephens criticised the Open University's use of the Pascal programming language in its introductory course, Fundamentals of Computing, and complained of what he saw as the dismissive attitude to business computing that is promoted by British educational establishments. Not surprisingly, this article provoked some fierce debate among PC PLUS readers. Here's a representative sample of the letters sent in response.

I really can't let Paul Stephens' article (December PC PLUS) on computer education pass without comment. The previous letters about *Pecan Pascal* did not inspire me to write, but Paul's did.

As usual, we have the tired old confusion between training and education. Whatever the reasons behind the Open University choosing *Pecan Pascal*, Paul seems to have missed the point that it is not there to teach students Pascal at all! Instead it teaches concepts, in particular those which underlie programming, software engineering and other aspects of computing. Having taken the course myself, I can assure you that there is much more to it than programming.

Education should not confuse students in the early stages by using systems which show exceptions, or which don't fit in with basic concepts. At the same time, it should illuminate theory with practical work. This means that when teaching basic programming concepts, one needs to provide students with a language which demands prior planning, can be written in a modular way, has strong typing, is fairly easy to read and is flexible.

Do C, *dBASE* and the like really meet these criteria? I think not, but Pascal does, and BASIC would if it had a bit more structure to it.

Of course, once you have learned about programming, then it should not be beyond your ability to move to C or *dBASE* or whatever. I have now written programs in these, and Forth, Assembler, Lisp as well. And of course, don't forget that the OU course referred to is a first introduction! Writing compilers is another way to grasp the underlying concepts. Useless for training I agree, but invaluable for education all the same.

Complaints about Pascal hardly come well from an industry which is notorious for wanting other people to do its training for it. When was the last time anyone asked for a trainee programmer in *dBASE*? Oh no! Two years' minimum experience before anyone will employ you, though how do you get the experience?

The solution to the problem of a lack of programmers in appropriate languages is simple, and in the hands of employers. If you want someone who is trained to do a particular job, then train them. British industry is famed for its unwillingness to train personnel, always wanting someone else to pick up the tab. But what Mr Stephens, and others in the industry, must realise is that universities and colleges are, like schools, there to educate. It is the employer's responsibility to train.

Incidentally, I work neither at the OU or in the IT industry. I teach IT at a secondary school (where programming is hardly stressed, though we do use BASIC and *dBASE* sometimes).
A Burrows
Wirral
Merseyside.

WRITE TO US

ELECTRONIC COMMENT

Paul's Pascal-bashing column sparked off a programming debate in the PC PLUS conference on the CIX electronic conferencing system. Here are a few messages.

Note – a couple of CIX shorthand terms:
IMHO – In My Humble Opinion
TPTB – The Powers That Be

```
============================
pcplus/general #97, from gmurray.

TITLE: Pascal as a Teaching Language

2 points on the Paul Stephens Article in Dec issue.

a) I think that being taught programming is more
important than the language used to do it. I would
consider any of the following languages suitable.
Pascal, C, Ada, Modula-2, Algol68. (Not an exclusive
list.)

b) The lack of built-in screen and disk handling is
IMHO an advantage. It gives the programmer the choice
of implementing whatever file structure or organisation
he/she considers to be best suited to the application.
Toolboxes are available for screen handling and common
file organisations. The programmer has the choice of
using one of these or not.

Graham

============================
pcplus/general #98, from graham
This is a comment to message 97

I have to say that I agreed totally with Paul
Stephens' article. I feel that your points are covered
by the article and that you are wrong. I cannot
believe your list of languages, which seem to be the
worst possible (except for PL/1 which I don't think
you included).

I would like to expand on that but feel that I would
just be rewriting the article.

============================
pcplus/general #99, from gmurray
This is a comment to message 98.

I still reckon that the important thing is knowing how
to go about solving the problem, finding the best
algorithm etc, rather than the implementation
language. In commercial applications, the first thing
that is done is to design the file and data structures.

To take a typical commercial application, an ordering
system. The basic record would consist of the customer
(or supplier) details followed by the details of the
items on the order. If this were written in COBOL the
record structure would be something like

    01 ORDERREC.
    05 SUPPLIERDETAILS.
    :
    :
    05 NUMBERITEMS PIC 99 COMP SYNC.
    05 ORDERLINES OCCURS 1 TO ? DEPENDING NUMBERITEMS.
    10 ITEMCODE PIC X(4).
    10 PRICE PIC 9(8) COMP SYNC.
    etc

where you have to decide the maximum number of lines
in an order. In memory space is allocated the maximum
number of items. Also in dBASE you have to decide
```
▶ 95

▶ 92   on the number of items, or make the
        application more complex by having 2 files, one
for Order details and another for items, referencing
the order details file.

By using any of the languages I mention, the number of
items can be dynamically determined, there is no
predetermined maximum number.

When I was working in a commercial environment, I
reckon that at least 25% of all program amendments were
caused by having to increase the number of occurrences
of a data item in some file or another.

Graham

============================
pcplus/general #100, from jonc
This is a comment to message 99.
_____

Can I just say that in COBOL I wouldn't (and don't)
define that sort of thing like that.

You started off all right by saying the the algorithm
(logical design) is of far more importance than the
implementation detail (let alone the implementation
language).

I have no objection to starting with the data
structures (NOTE: not FILE structures) but in an
abstract way, worring about how that is implementated
(files, hierarchical DB, relational DB OODBMS etc) comes
later (physical design) and yes I would agree is
influenced by implementation tools available.

The advantage of Pascal (bog standard ISO pascal that
is) is you can teach the concepts of structure, flow
control, encapsulation etc without resorting to what is
always platform specific stuff like Input/Output
(especially user I/O). I will admit that any real
language will have to extend that but that is not
particularly relevant in the early stages of teaching.

============================
pcplus/general #101, from terrym
This is a comment to message 100.
_____

I agree that Pascal is the best 'teaching' language. C
is a little terse and BASIC - well.......

============================
pcplus/general #102, from nleverton
This is a comment to message 99.
There are comments to this message.
_____

So the designs were wrong, weren't they! I wouldn't use
a fixed size table for something as dynamic as number of
lines on an order. Only for something like VAT where
you can be 99% sure TPTB won't use more than five rates
(leastways they *said* they won't ...)

I do agree that learning to design good algorithms and
structures is much more important than learning in any
particular language. IMHO you either can [learn to]
program or you can't. I've seen an awful lot of people
who just had not got the mindset for understanding even
the most simple application, although some of them had
been coding for years and were pretty knowledgeable
about, say, COBOL.

============================
pcplus/general #103, from gale
This is a comment to message 102.
_____

I have to agree with your sentiments.
In my proper job (i.e., what I do when not logged in to

Cix), I work with the ICL VME Operating System wherein
absolutely everything, including code and data
segmentation, is implemented as the equivalent of
stacks, with all the 'infinite' open-endedness inherent
in stack-type structures.

It infuriates me to find, as I frequently do, that
programmers or software system designers build false
limits into their designs which are imposed by neither
the hardware nor the operating system.

Typically mindless statements from such people are:
    - no-one will ever want to order more than fifty
items at once;
    - no program could possibly want to process more
than forty files
    concurrently;
    - the chances of two users trying to access the same
database record
    at the same time are negligible;
    - this program won't be in use for more than five
years so just
    cater for years beginning with '19'.

Ooh, it gets my dander right up it does!

Gale.

THE LAST WORD...

*Finally, the view of the Open University itself, expressed not to us but
to a prospective student. Chris Sandall of Milton Keynes (home, as it
happens, of the OU) was about to enroll on the Fundamentals of
Computing course when he read Paul's comments. He wrote to the
University, enclosing a copy of the article and asking for their
assurances that the course would be 'relevant, worthwhile and lead
to a job in computing. Here's the reply Chris received.*

Dear Mr. Sandall,
I have been asked to comment on the Paul Stephens article, and the
suitability of M205 *(the Fundamentals of Computing course – Ed)* for
somebody like yourself.
    We are currently planning M206, a rewrite of M205 for 1994.
One question we must resolve concerns which language to use. To
help us, we have surveyed many of the country's Universities and
Polytechnics to find out what they are using. With one or two
exceptions, they are all using either Pascal or Modula 2 – a language
which developed out of Pascal – as the language to be first taught. I
hope this sets your mind at rest.
    As to Stephens' recommendation of *dBASE*, I cannot believe he is
serious. That would concentrate on one, narrow application of
computers, whereas M205 is a broad introduction to the whole
software side of Computer Science.
Yours Sincerely,
**L. A. Best,
M205 E and A Board Chairman.**

'Stephens' comments:
*Chris Sandall specifically asked if the course would lead to a job in
computing; the OU chose to answer this question by stating that
most universities and polytechnics are using Pascal and Modula.
This, it was suggested, should set his mind at rest. It then described
business data processing (the area which dBASE is designed for) as
'one, narrow, application of computers'.*
    *In one respect, this reply can indeed set Chris's mind at rest – he
can always get a job at a university or polytechnic, teaching Pascal
and Modula! That would be typical of the 'closed circuit' process in
which a student becomes a teacher without gaining any practical
experience. It protects the academic world from intrusion by
outsiders laden down with truths about what it should really be
providing for its students. It's precisely the kind of attitude shown in
this letter which is the cause of so many of the problems in our
higher education system.*